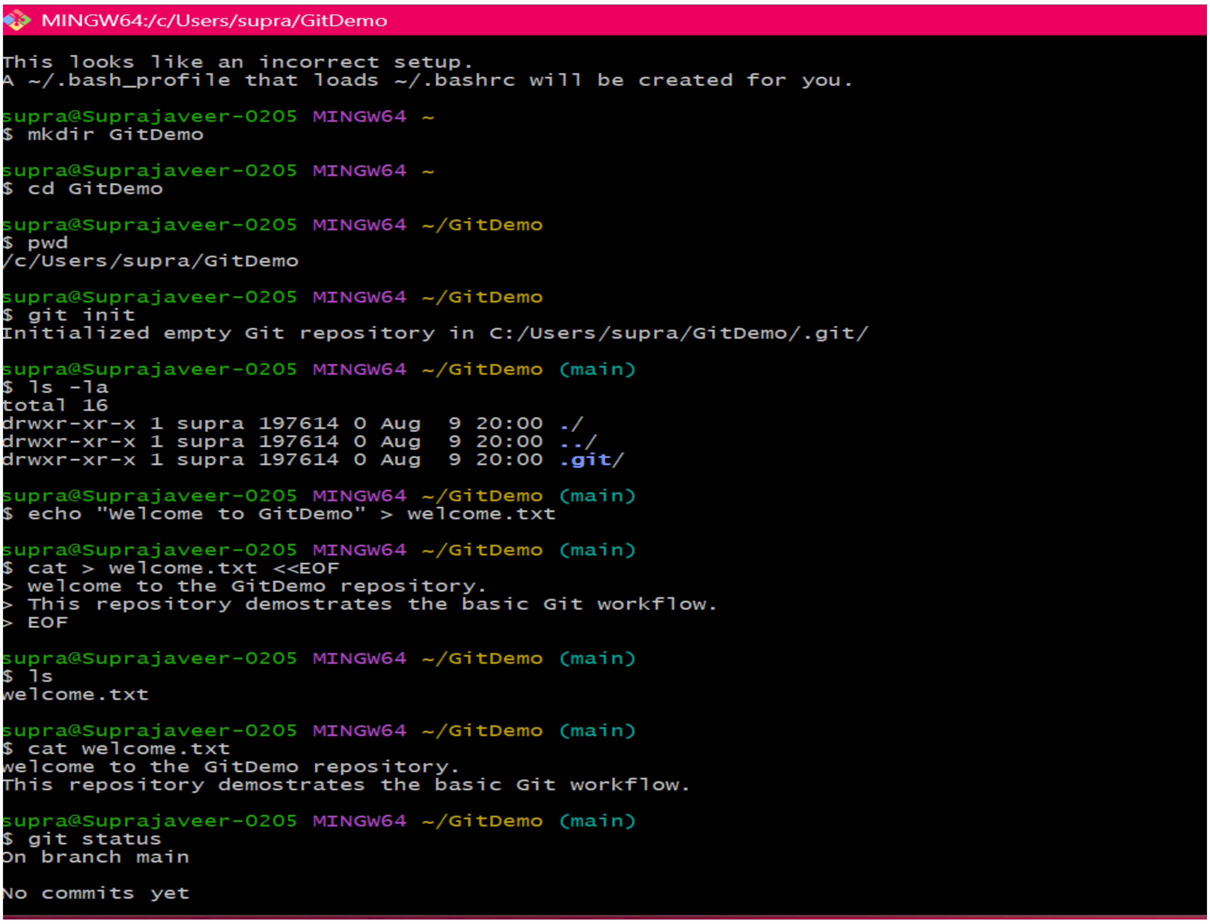
WEEK-8 GIT

1.GIT-HOL

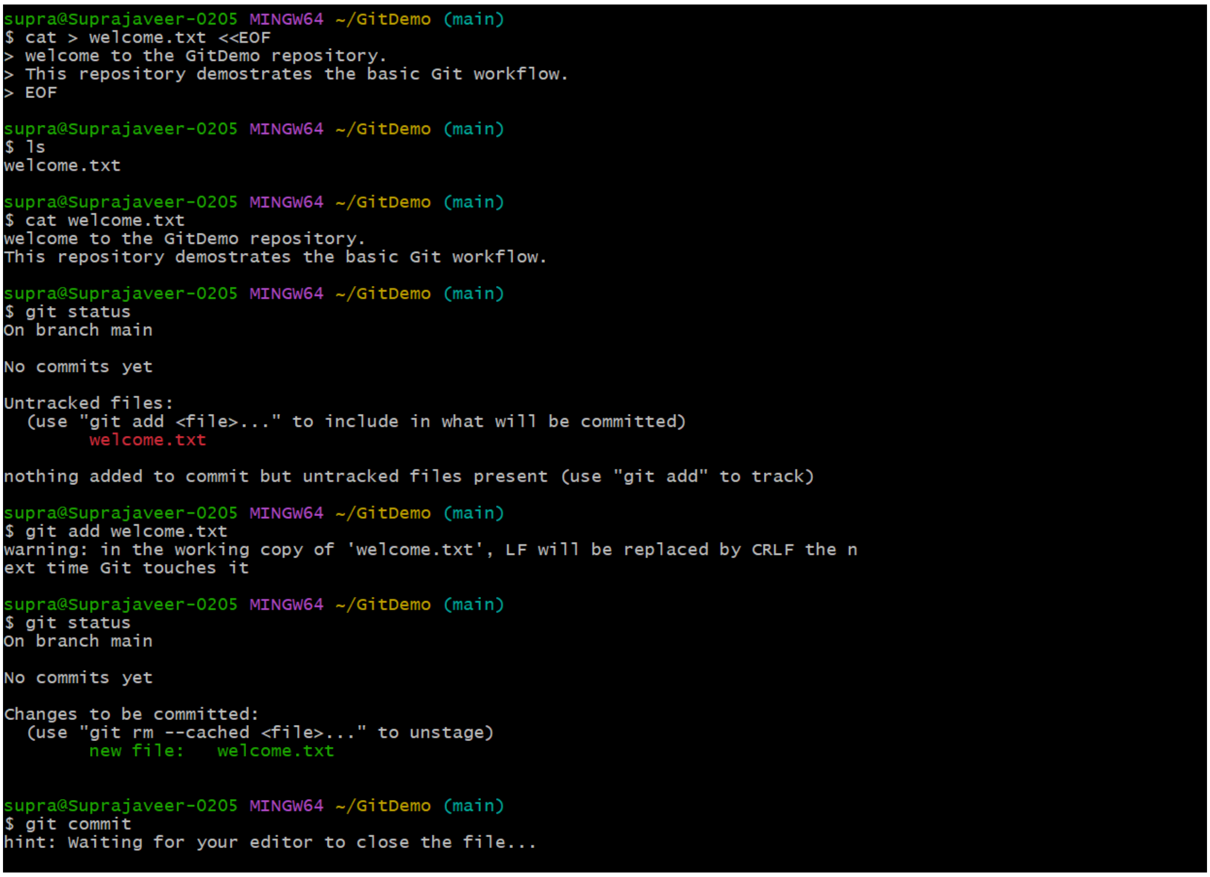
To create a new repository, signup with GitLab and register your credentials

Login to GitLab and create a “GitDemo” project

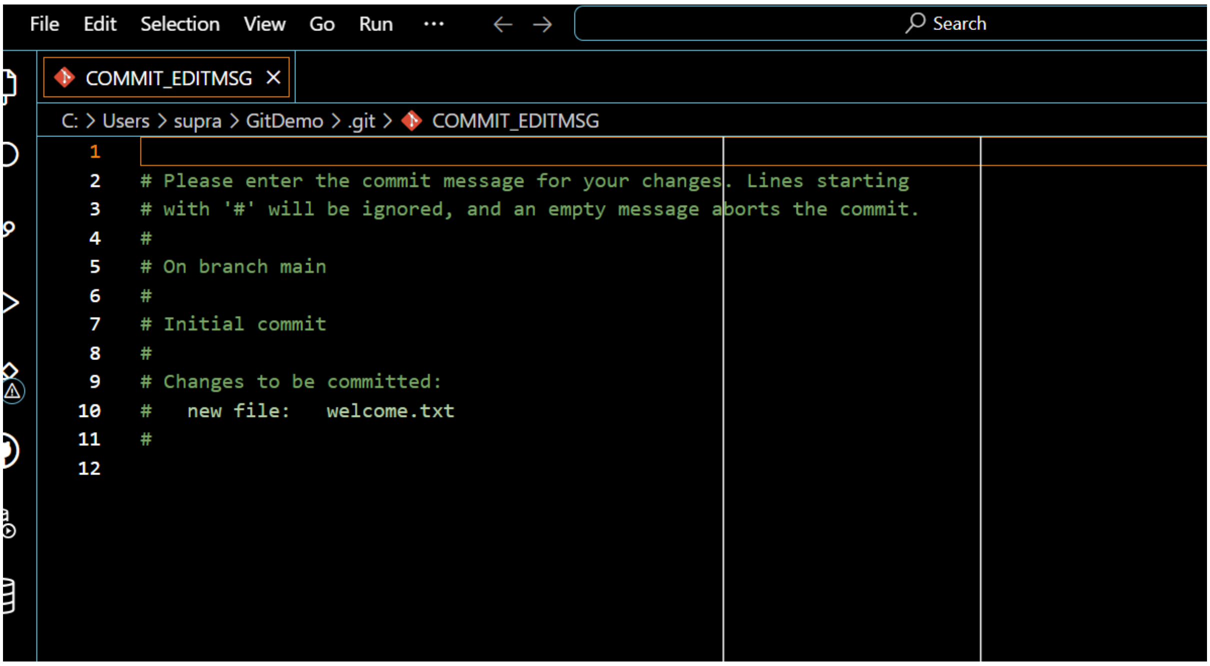
OUTPUT:



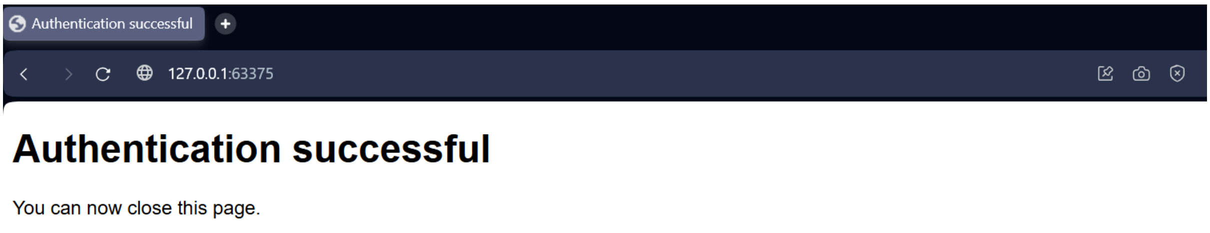
```
This looks like an incorrect setup.
A ~/.bash_profile that loads ~/.bashrc will be created for you.

supra@Suprajaveer-0205 MINGW64 ~
$ mkdir GitDemo

supra@Suprajaveer-0205 MINGW64 ~
$ cd GitDemo

supra@Suprajaveer-0205 MINGW64 ~/GitDemo
$ pwd
/c/Users/supra/GitDemo

supra@Suprajaveer-0205 MINGW64 ~/GitDemo
$ git init
Initialized empty Git repository in C:/Users/supra/GitDemo/.git/

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ ls -la
total 16
drwxr-xr-x 1 supra 197614 0 Aug  9 20:00 ./
drwxr-xr-x 1 supra 197614 0 Aug  9 20:00 ../
drwxr-xr-x 1 supra 197614 0 Aug  9 20:00 .git/

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ echo "Welcome to GitDemo" > welcome.txt

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ cat > welcome.txt <<EOF
> welcome to the GitDemo repository.
> This repository demostrates the basic Git workflow.
> EOF

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ ls
welcome.txt

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ cat welcome.txt
welcome to the GitDemo repository.
This repository demostrates the basic Git workflow.

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git status
On branch main

No commits yet
```



```
supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ cat > welcome.txt <<EOF
> welcome to the GitDemo repository.
> This repository demostrates the basic Git workflow.
> EOF

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ ls
welcome.txt

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ cat welcome.txt
welcome to the GitDemo repository.
This repository demostrates the basic Git workflow.

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        welcome.txt

nothing added to commit but untracked files present (use "git add" to track)

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git add welcome.txt
warning: in the working copy of 'welcome.txt', LF will be replaced by CRLF the n
ext time Git touches it

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   welcome.txt


supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git commit
hint: Waiting for your editor to close the file...
```

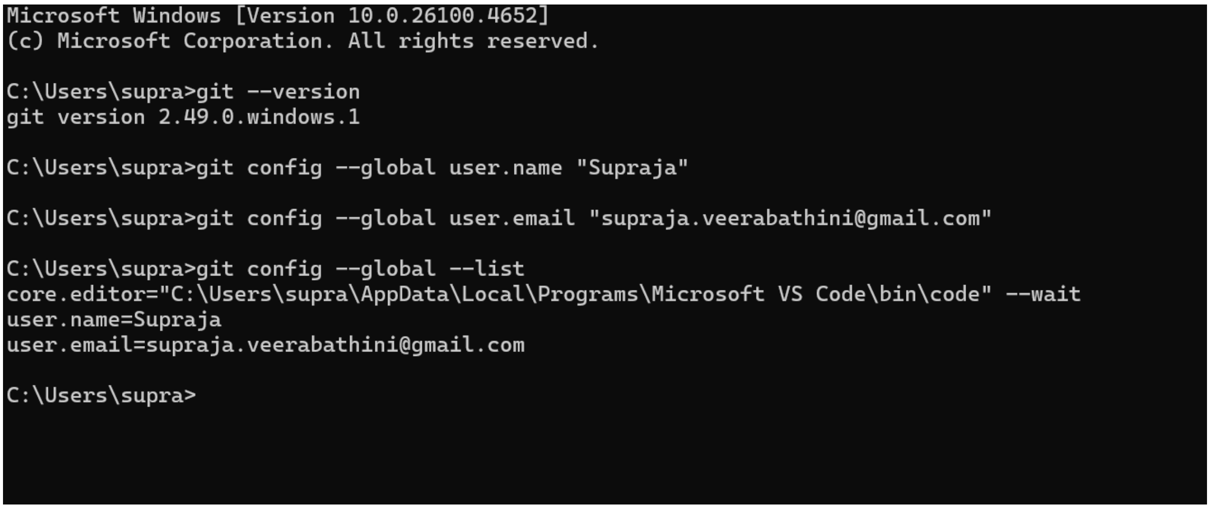


```

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch main
#
# Initial commit
#
# Changes to be committed:
#	new file:   welcome.txt
#

```

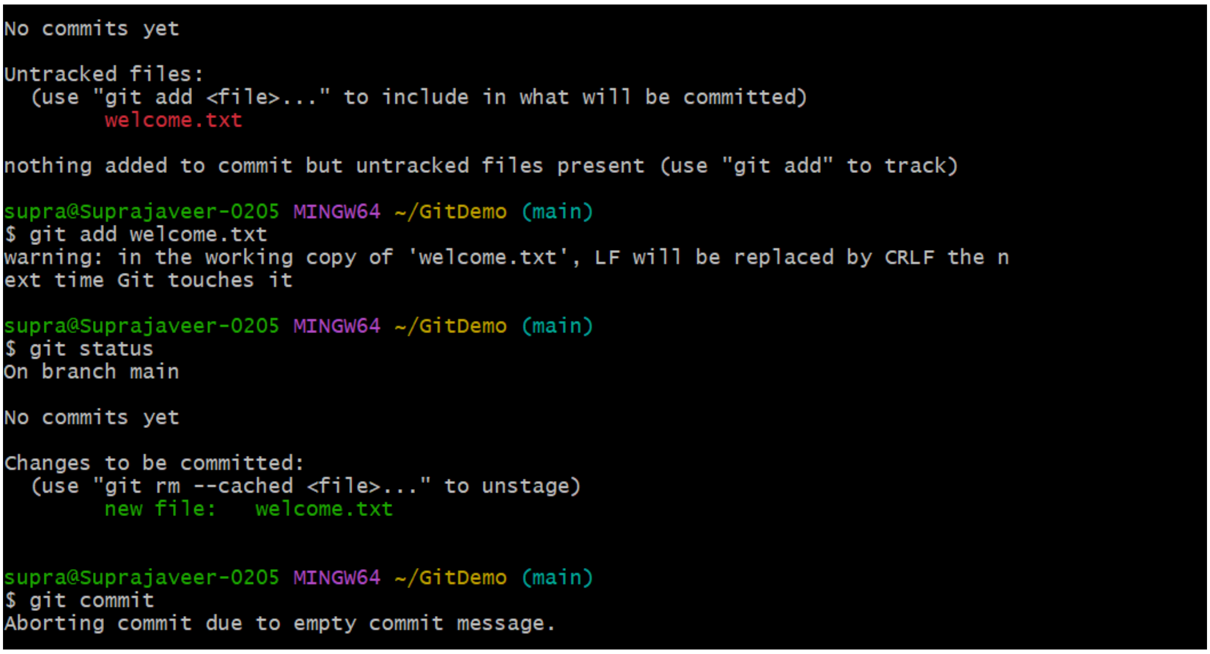


Authentication successful

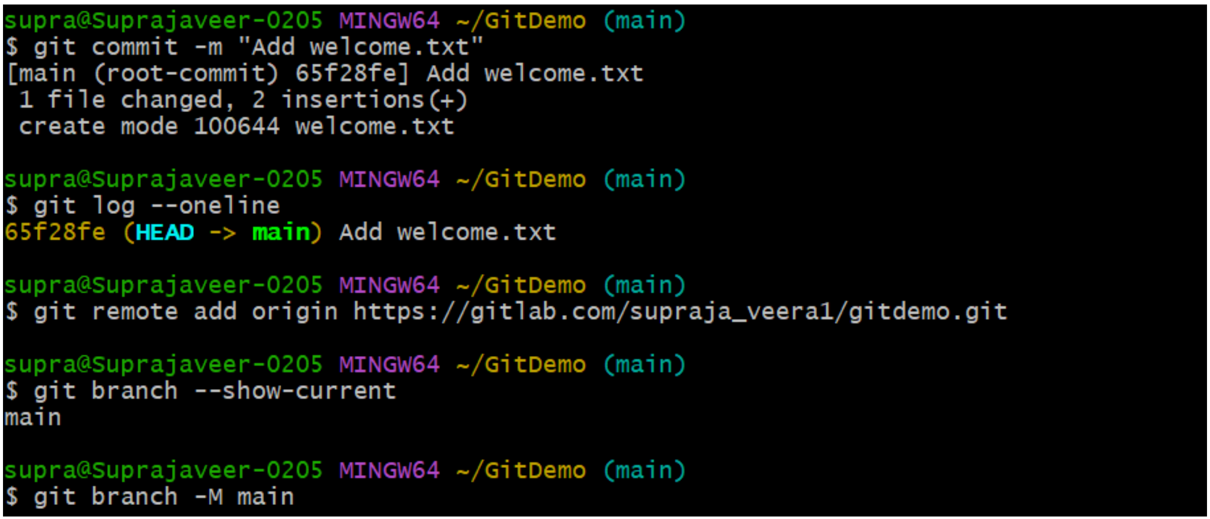
You can now close this page.



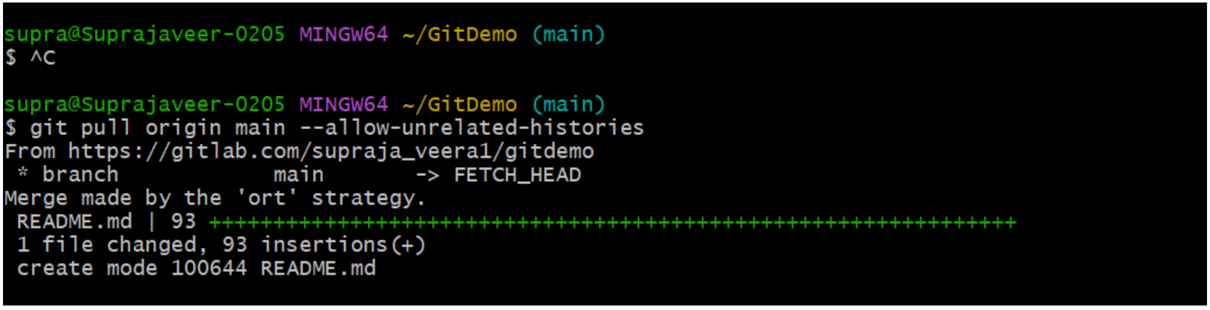
```
Microsoft Windows [Version 10.0.26100.4652]
(c) Microsoft Corporation. All rights reserved.

C:\Users\supra>git --version
git version 2.49.0.windows.1

C:\Users\supra>git config --global user.name "Supraja"

C:\Users\supra>git config --global user.email "supraja.veerabathini@gmail.com"

C:\Users\supra>git config --global --list
core.editor="C:\Users\supra\AppData\Local\Programs\Microsoft VS Code\bin\code" --wait
user.name=Supraja
user.email=supraja.veerabathini@gmail.com

C:\Users\supra>
```



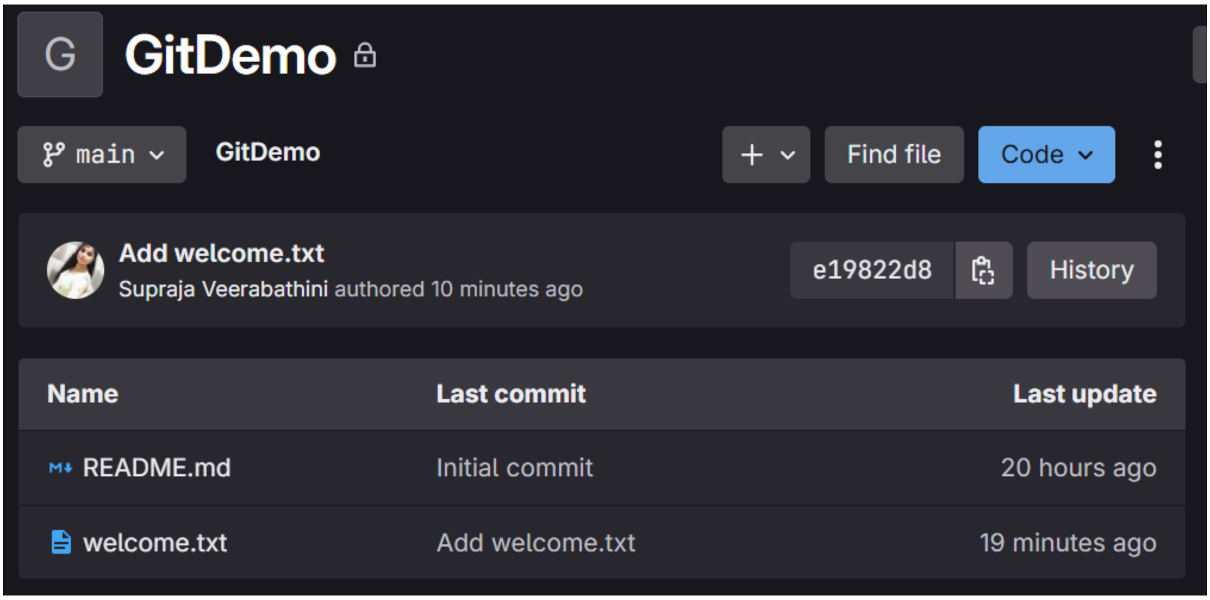
```
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        welcome.txt

nothing added to commit but untracked files present (use "git add" to track)

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git add welcome.txt
warning: in the working copy of 'welcome.txt', LF will be replaced by CRLF the n
ext time Git touches it

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   welcome.txt


supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git commit
Aborting commit due to empty commit message.
```



```
supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git commit -m "Add welcome.txt"
[main (root-commit) 65f28fe] Add welcome.txt
 1 file changed, 2 insertions(+)
 create mode 100644 welcome.txt

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git log --oneline
65f28fe (HEAD -> main) Add welcome.txt

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git remote add origin https://gitlab.com/supraja_veera1/gitdemo.git

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git branch --show-current
main

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git branch -M main
```



```
supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ ^C

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git pull origin main --allow-unrelated-histories
From https://gitlab.com/supraja_veera1/gitdemo
 * branch            main       -> FETCH_HEAD
Merge made by the 'ort' strategy.
 README.md | 93 +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 93 insertions(+)
 create mode 100644 README.md
```



Add welcome.txt
Supraja Veerabathini authored 10 minutes ago

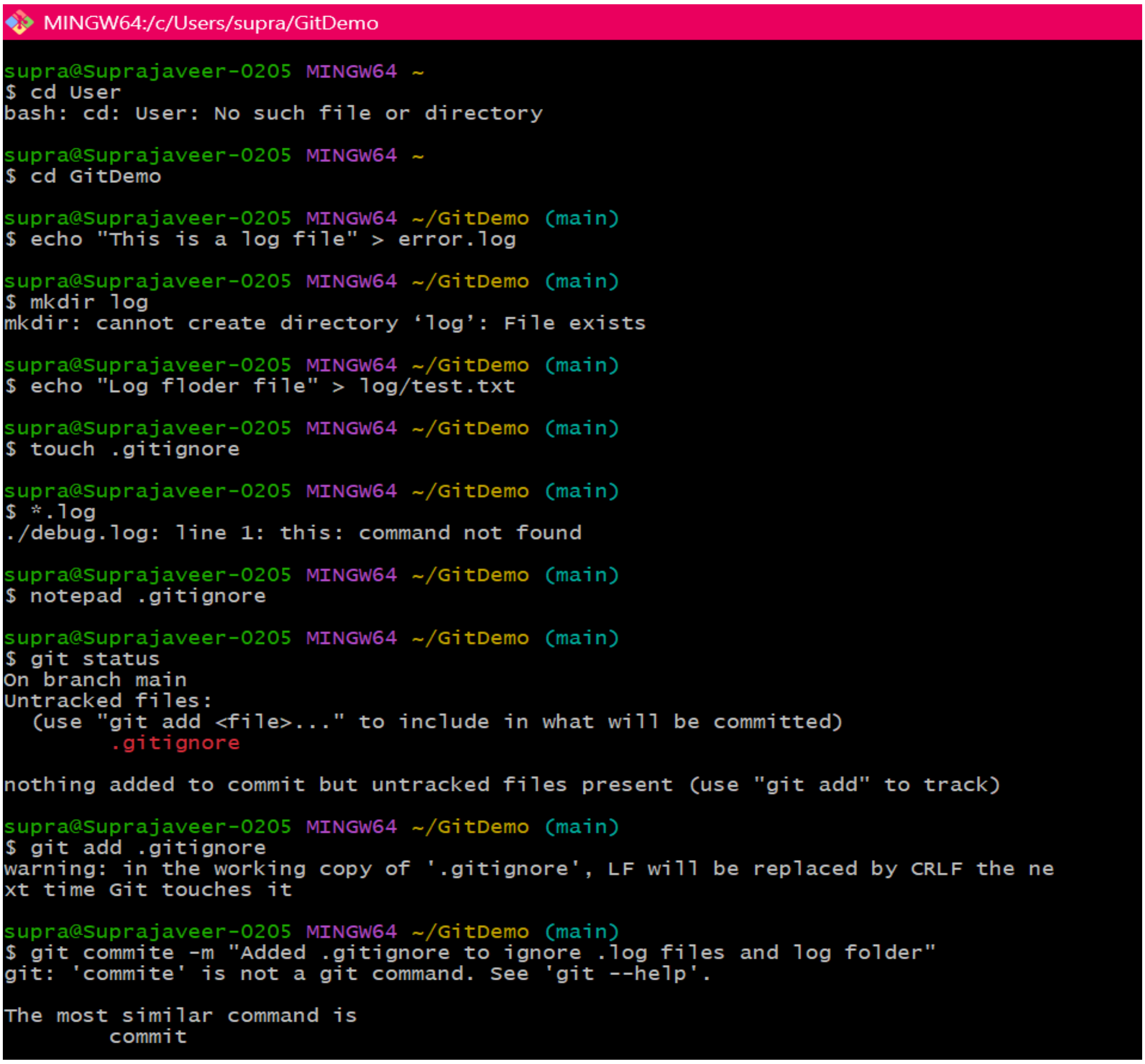
Name	Last commit	Last update
README.md	Initial commit	20 hours ago
welcome.txt	Add welcome.txt	19 minutes ago

2.GIT-HOL

Create a **".log"** file and a **log folder** in the working directory of Git. Update the **.gitignore** file in such a way that on committing, these files (.log extensions and log folders) are ignored.

Verify if the git status reflects the same about working directory, local repository and git repository.

OUTPUT:



```
supra@Suprajaveer-0205 MINGW64 ~
$ cd User
bash: cd: User: No such file or directory

supra@Suprajaveer-0205 MINGW64 ~
$ cd GitDemo

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ echo "This is a log file" > error.log

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ mkdir log
mkdir: cannot create directory ‘log’: File exists

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ echo "Log floder file" > log/test.txt

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ touch .gitignore

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ *.log
./debug.log: line 1: this: command not found

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ notepad .gitignore

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore

nothing added to commit but untracked files present (use "git add" to track)

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git add .gitignore
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the ne
xt time Git touches it

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git commite -m "Added .gitignore to ignore .log files and log folder"
git: 'commite' is not a git command. See 'git --help'.

The most similar command is
        commit
```

3.GIT-HOL

Branching:

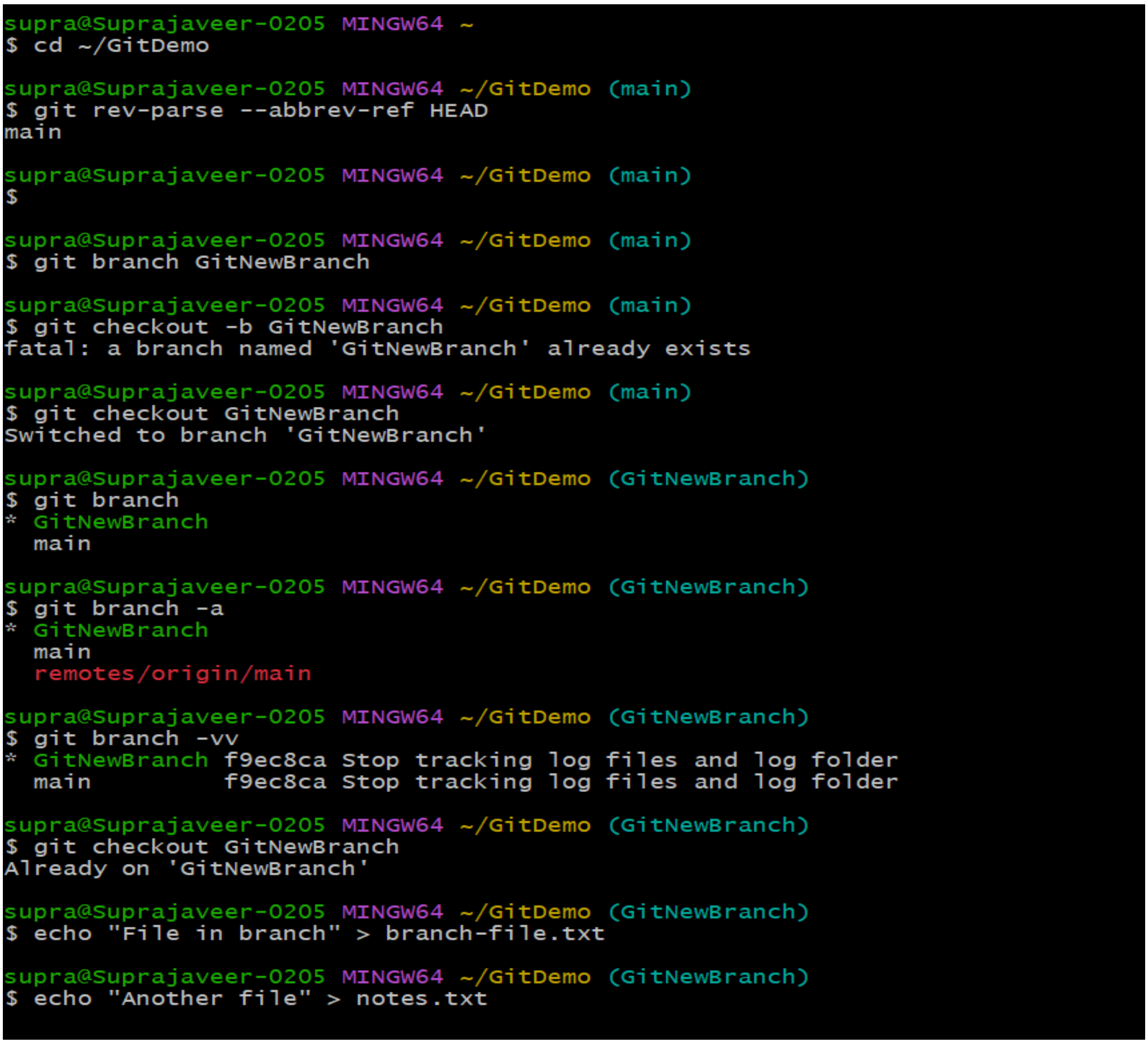
1. Create a new branch **“GitNewBranch”.**
2. List all the local and remote branches available in the current trunk. Observe the “*” mark which denote the current pointing branch.
3. Switch to the newly created branch. Add some files to it with some contents.
4. Commit the changes to the branch.
5. Check the status with **“git status”** command.

Merging:

1. Switch to the master
2. List out all the differences between trunk and branch. These provide the differences in command line interface.

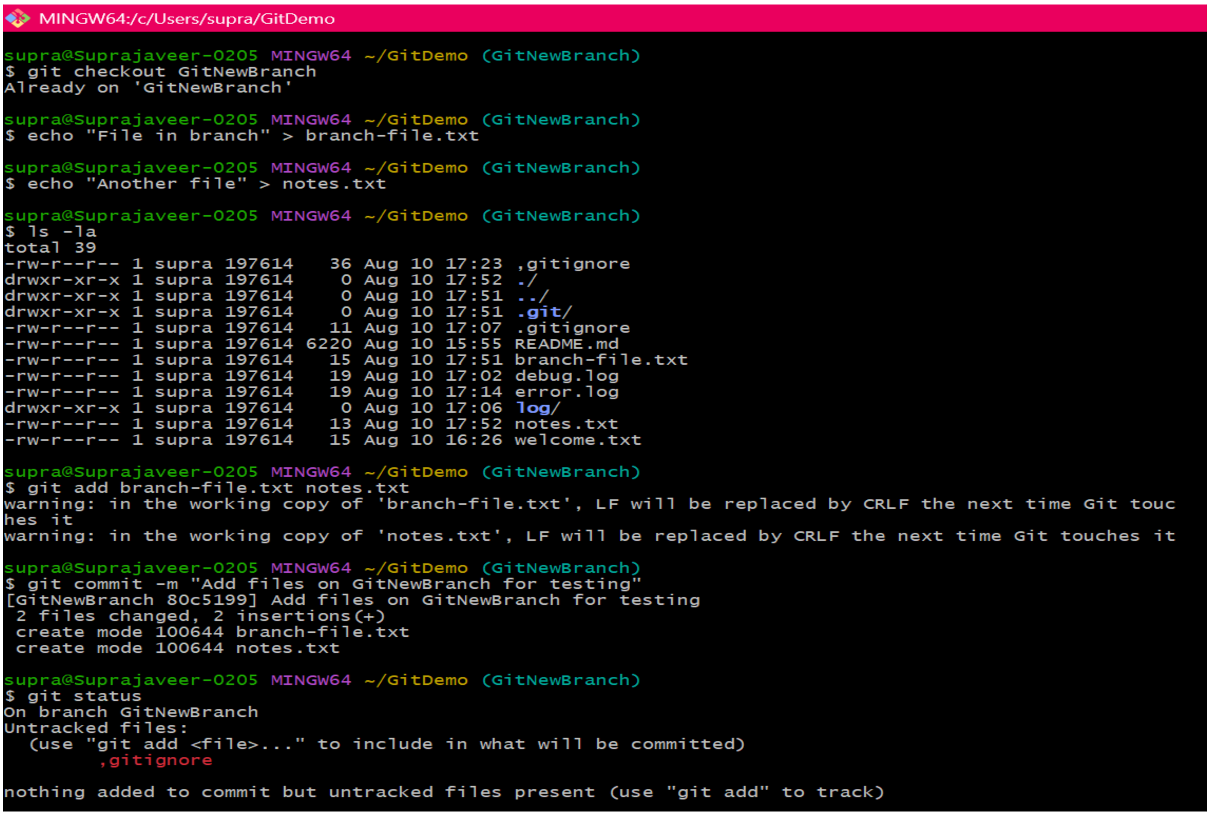
3. List out all the visual differences between master and branch using **P4Merge tool**.
4. Merge the source branch to the trunk.
5. Observe the logging after merging using **“git log –oneline –graph –decorate”**
6. Delete the branch after merging with the trunk and observe the git status

OUTPUT:



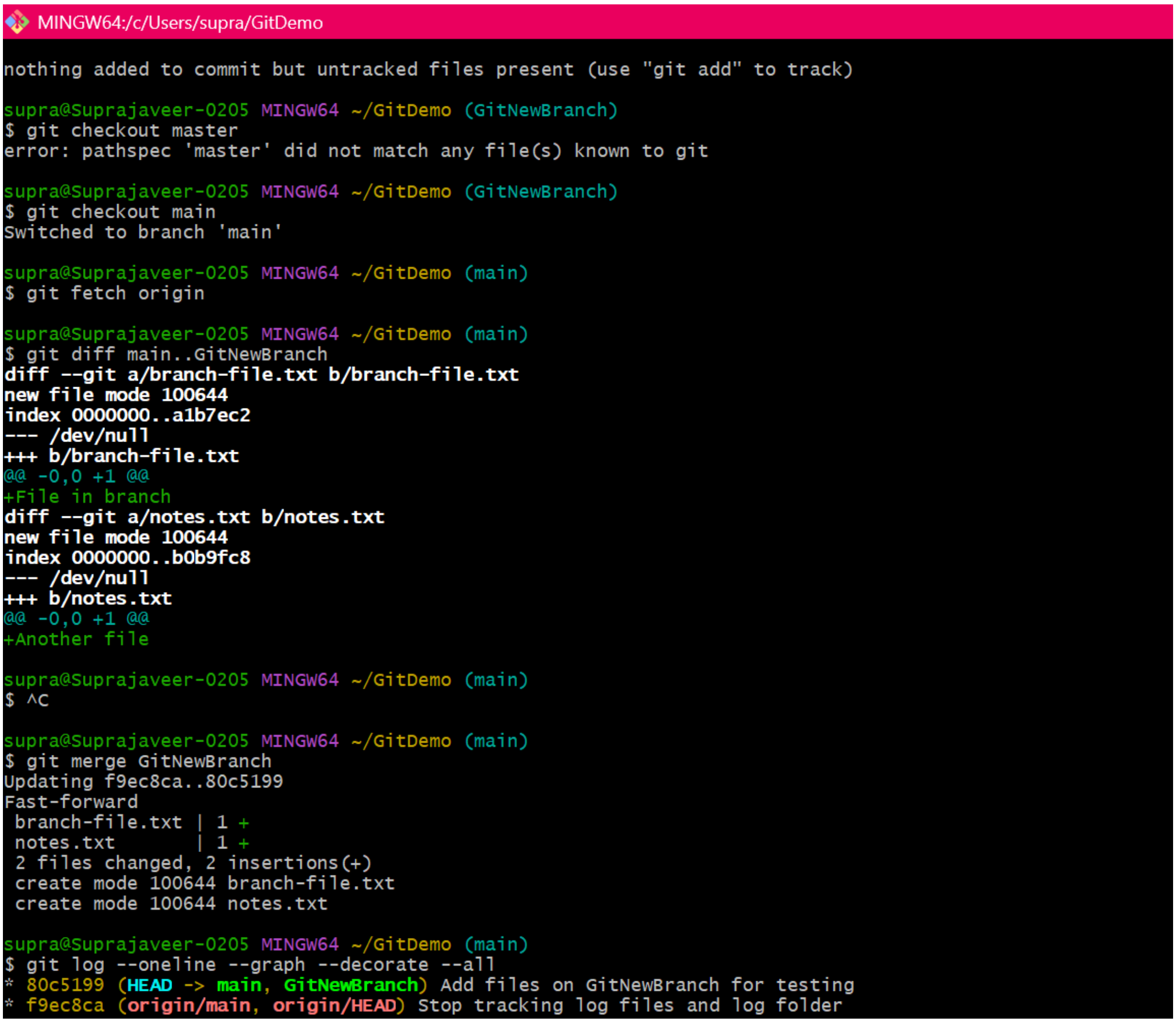
```
supra@Suprajaveer-0205 MINGW64 ~
$ cd ~/GitDemo

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git rev-parse --abbrev-ref HEAD
main

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git branch GitNewBranch

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git checkout -b GitNewBranch
fatal: a branch named 'GitNewBranch' already exists

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git checkout GitNewBranch
Switched to branch 'GitNewBranch'

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (GitNewBranch)
$ git branch
* GitNewBranch
  main

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (GitNewBranch)
$ git branch -a
* GitNewBranch
  main
  remotes/origin/main

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (GitNewBranch)
$ git branch -vv
* GitNewBranch f9ec8ca Stop tracking log files and log folder
  main         f9ec8ca Stop tracking log files and log folder

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (GitNewBranch)
$ git checkout GitNewBranch
Already on 'GitNewBranch'

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (GitNewBranch)
$ echo "File in branch" > branch-file.txt

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (GitNewBranch)
$ echo "Another file" > notes.txt
```

.

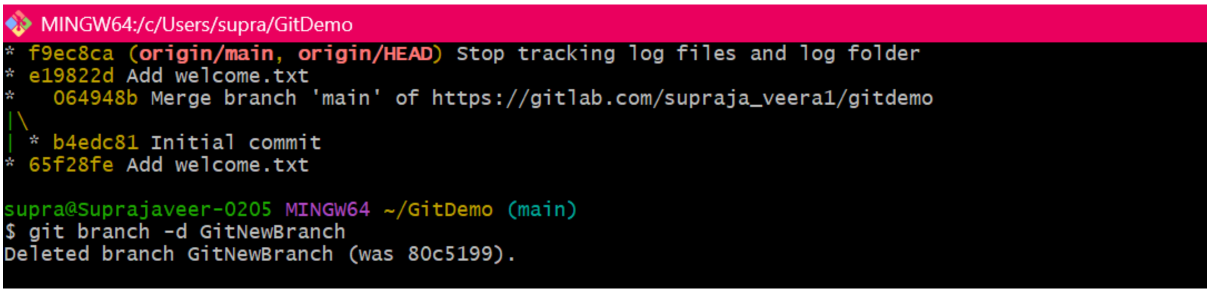


MINGW64:/c/Users/supra/GitDemo

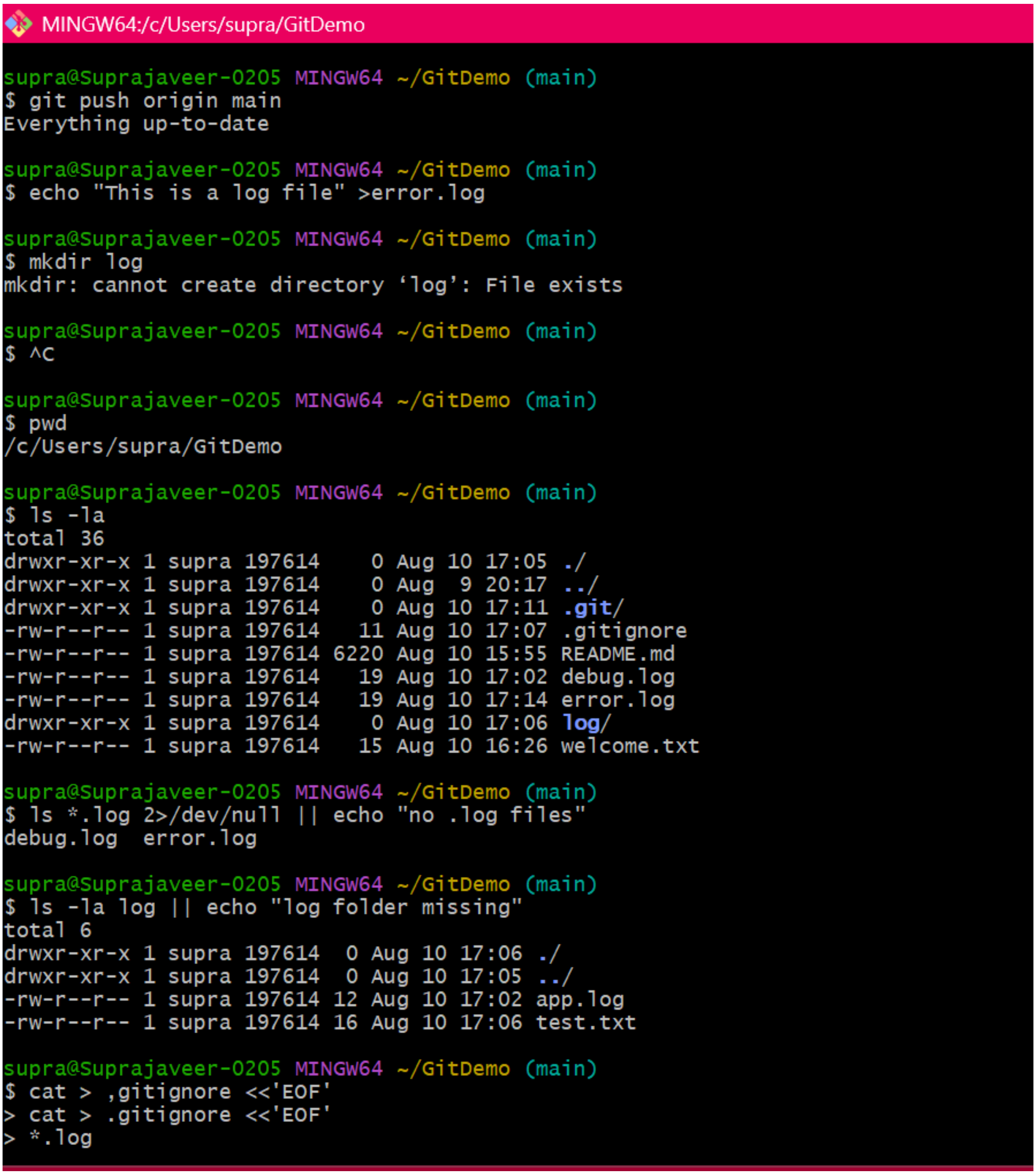
```
supra@Suprajaveer-0205 MINGW64 ~/GitDemo (GitNewBranch)
$ git checkout GitNewBranch
Already on 'GitNewBranch'

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (GitNewBranch)
$ echo "File in branch" > branch-file.txt

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (GitNewBranch)
$ echo "Another file" > notes.txt

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (GitNewBranch)
$ ls -la
total 39
-rw-r--r-- 1 supra 197614   36 Aug 10 17:23 ,gitignore
drwxr-xr-x 1 supra 197614    0 Aug 10 17:52 ./
drwxr-xr-x 1 supra 197614    0 Aug 10 17:51 ../
drwxr-xr-x 1 supra 197614    0 Aug 10 17:51 .git/
-rw-r--r-- 1 supra 197614   11 Aug 10 17:07 .gitignore
-rw-r--r-- 1 supra 197614 6220 Aug 10 15:55 README.md
-rw-r--r-- 1 supra 197614   15 Aug 10 17:51 branch-file.txt
-rw-r--r-- 1 supra 197614   19 Aug 10 17:02 debug.log
-rw-r--r-- 1 supra 197614   19 Aug 10 17:14 error.log
drwxr-xr-x 1 supra 197614    0 Aug 10 17:06 log/
-rw-r--r-- 1 supra 197614   13 Aug 10 17:52 notes.txt
-rw-r--r-- 1 supra 197614   15 Aug 10 16:26 welcome.txt

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (GitNewBranch)
$ git add branch-file.txt notes.txt
warning: in the working copy of 'branch-file.txt', LF will be replaced by CRLF the next time Git touc
hes it
warning: in the working copy of 'notes.txt', LF will be replaced by CRLF the next time Git touches it

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (GitNewBranch)
$ git commit -m "Add files on GitNewBranch for testing"
[GitNewBranch 80c5199] Add files on GitNewBranch for testing
 2 files changed, 2 insertions(+)
 create mode 100644 branch-file.txt
 create mode 100644 notes.txt

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (GitNewBranch)
$ git status
On branch GitNewBranch
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ,gitignore

nothing added to commit but untracked files present (use "git add" to track)
```



```
nothing added to commit but untracked files present (use "git add" to track)

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (GitNewBranch)
$ git checkout master
error: pathspec 'master' did not match any file(s) known to git

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (GitNewBranch)
$ git checkout main
Switched to branch 'main'

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git fetch origin

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git diff main..GitNewBranch
diff --git a/branch-file.txt b/branch-file.txt
new file mode 100644
index 0000000..a1b7ec2
--- /dev/null
+++ b/branch-file.txt
@@ -0,0 +1 @@
+File in branch
diff --git a/notes.txt b/notes.txt
new file mode 100644
index 0000000..b0b9fc8
--- /dev/null
+++ b/notes.txt
@@ -0,0 +1 @@
+Another file

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ ^C

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git merge GitNewBranch
Updating f9ec8ca..80c5199
Fast-forward
 branch-file.txt | 1 +
 notes.txt       | 1 +
 2 files changed, 2 insertions(+)
 create mode 100644 branch-file.txt
 create mode 100644 notes.txt

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git log --oneline --graph --decorate --all
* 80c5199 (HEAD -> main, GitNewBranch) Add files on GitNewBranch for testing
* f9ec8ca (origin/main, origin/HEAD) Stop tracking log files and log folder
```

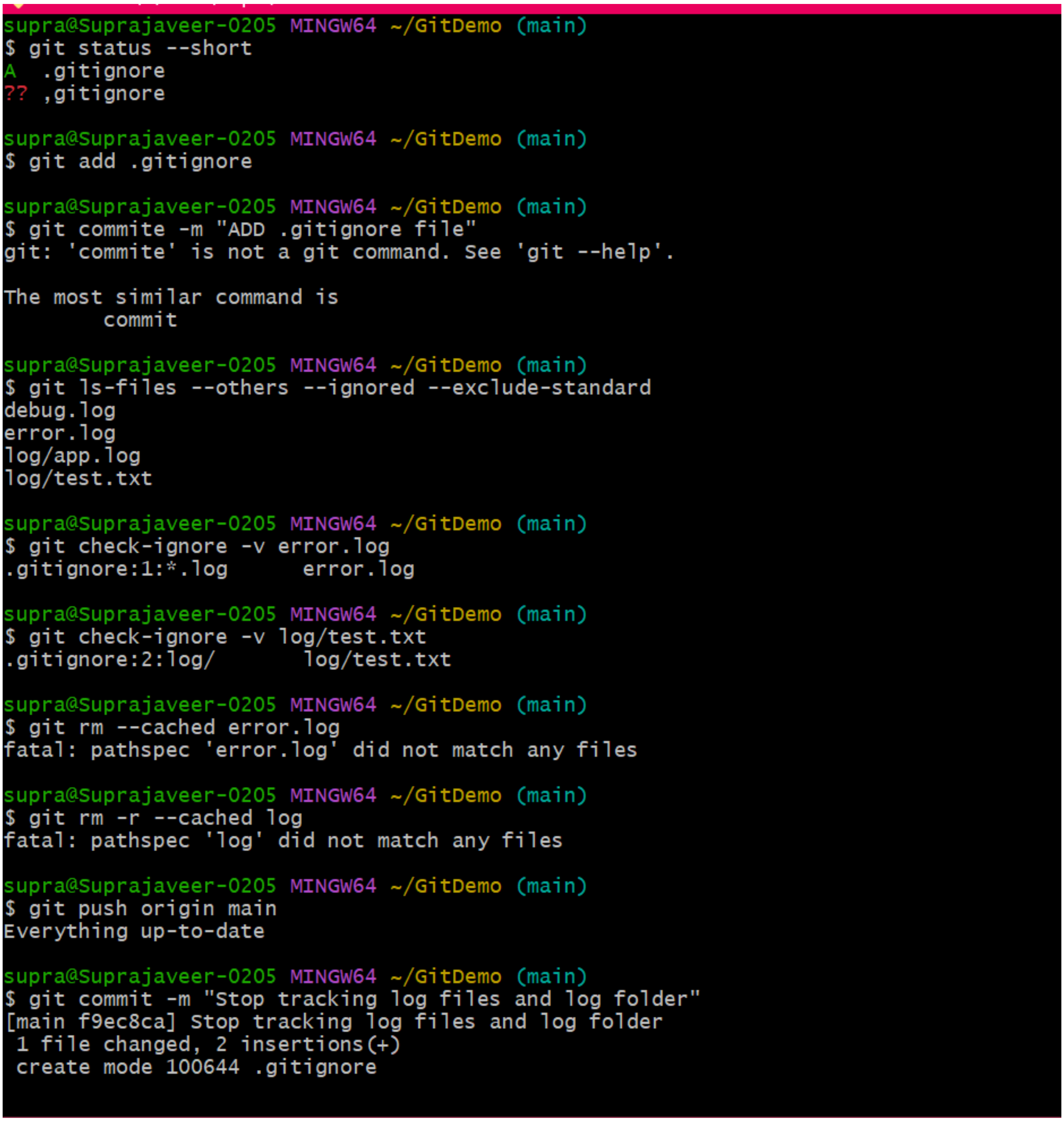


```
* f9ec8ca (origin/main, origin/HEAD) Stop tracking log files and log folder
* e19822d Add welcome.txt
*   064948b Merge branch 'main' of https://gitlab.com/supraja_veera1/gitdemo
|\
| * b4edc81 Initial commit
* 65f28fe Add welcome.txt

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git branch -d GitNewBranch
Deleted branch GitNewBranch (was 80c5199).
```

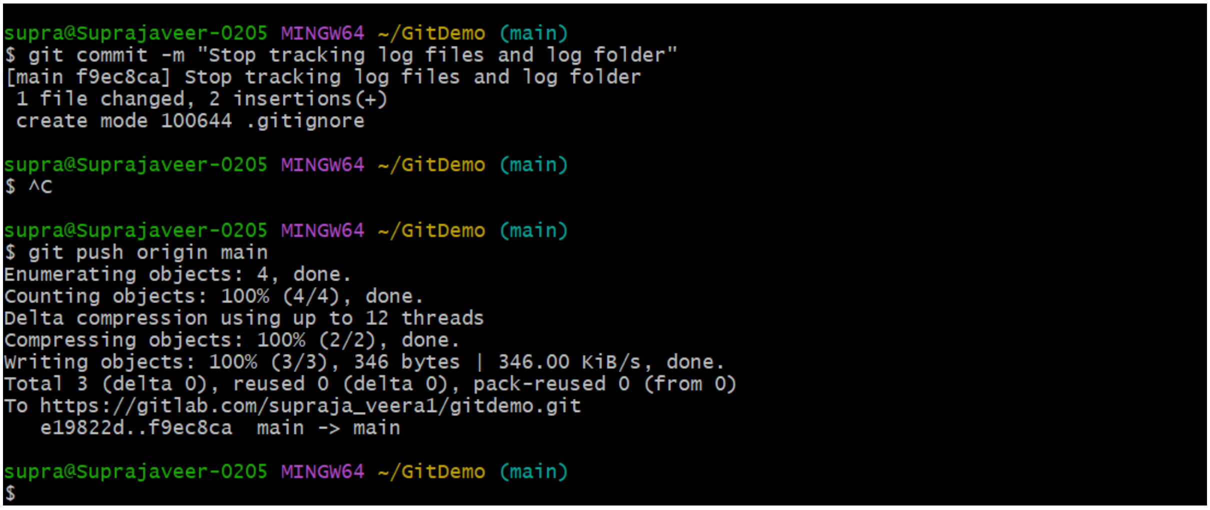


MINGW64:/c/Users/supra/GitDemo

```
supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git push origin main
Everything up-to-date

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ echo "This is a log file" >error.log

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ mkdir log
mkdir: cannot create directory ‘log’: File exists

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ ^C

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ pwd
/c/Users/supra/GitDemo

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ ls -la
total 36
drwxr-xr-x 1 supra 197614    0 Aug 10 17:05 ./
drwxr-xr-x 1 supra 197614    0 Aug  9 20:17 ../
drwxr-xr-x 1 supra 197614    0 Aug 10 17:11 .git/
-rw-r--r-- 1 supra 197614   11 Aug 10 17:07 .gitignore
-rw-r--r-- 1 supra 197614 6220 Aug 10 15:55 README.md
-rw-r--r-- 1 supra 197614   19 Aug 10 17:02 debug.log
-rw-r--r-- 1 supra 197614   19 Aug 10 17:14 error.log
drwxr-xr-x 1 supra 197614    0 Aug 10 17:06 log/
-rw-r--r-- 1 supra 197614   15 Aug 10 16:26 welcome.txt

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ ls *.log 2>/dev/null || echo "no .log files"
debug.log  error.log

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ ls -la log || echo "log folder missing"
total 6
drwxr-xr-x 1 supra 197614  0 Aug 10 17:06 ./
drwxr-xr-x 1 supra 197614  0 Aug 10 17:05 ../
-rw-r--r-- 1 supra 197614 12 Aug 10 17:02 app.log
-rw-r--r-- 1 supra 197614 16 Aug 10 17:06 test.txt

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ cat > ,gitignore <<'EOF'
> cat > .gitignore <<'EOF'
> *.log
```



```
supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git status --short
A  .gitignore
?? ,gitignore

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git add .gitignore

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git commite -m "ADD .gitignore file"
git: 'commite' is not a git command. See 'git --help'.

The most similar command is
        commit

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git ls-files --others --ignored --exclude-standard
debug.log
error.log
log/app.log
log/test.txt

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git check-ignore -v error.log
.gitignore:1:*.log      error.log

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git check-ignore -v log/test.txt
.gitignore:2:log/       log/test.txt

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git rm --cached error.log
fatal: pathspec 'error.log' did not match any files

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git rm -r --cached log
fatal: pathspec 'log' did not match any files

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git push origin main
Everything up-to-date

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git commit -m "Stop tracking log files and log folder"
[main f9ec8ca] Stop tracking log files and log folder
 1 file changed, 2 insertions(+)
 create mode 100644 .gitignore
```

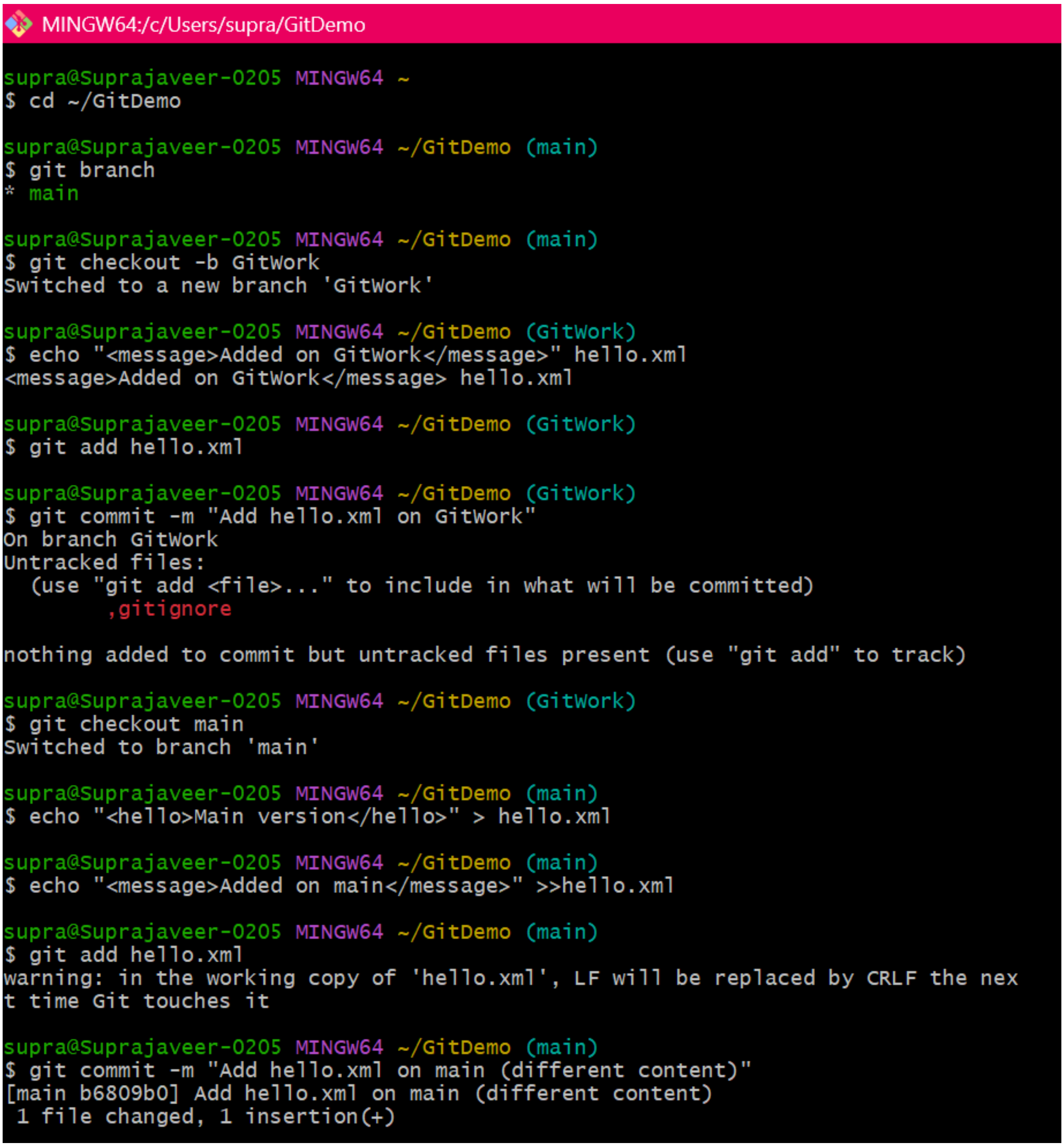


```
supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git commit -m "Stop tracking log files and log folder"
[main f9ec8ca] Stop tracking log files and log folder
 1 file changed, 2 insertions(+)
 create mode 100644 .gitignore

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ ^C

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 346 bytes | 346.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://gitlab.com/supraja_veeral/gitdemo.git
   e19822d..f9ec8ca  main -> main

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$
```

4.GIT-HOL

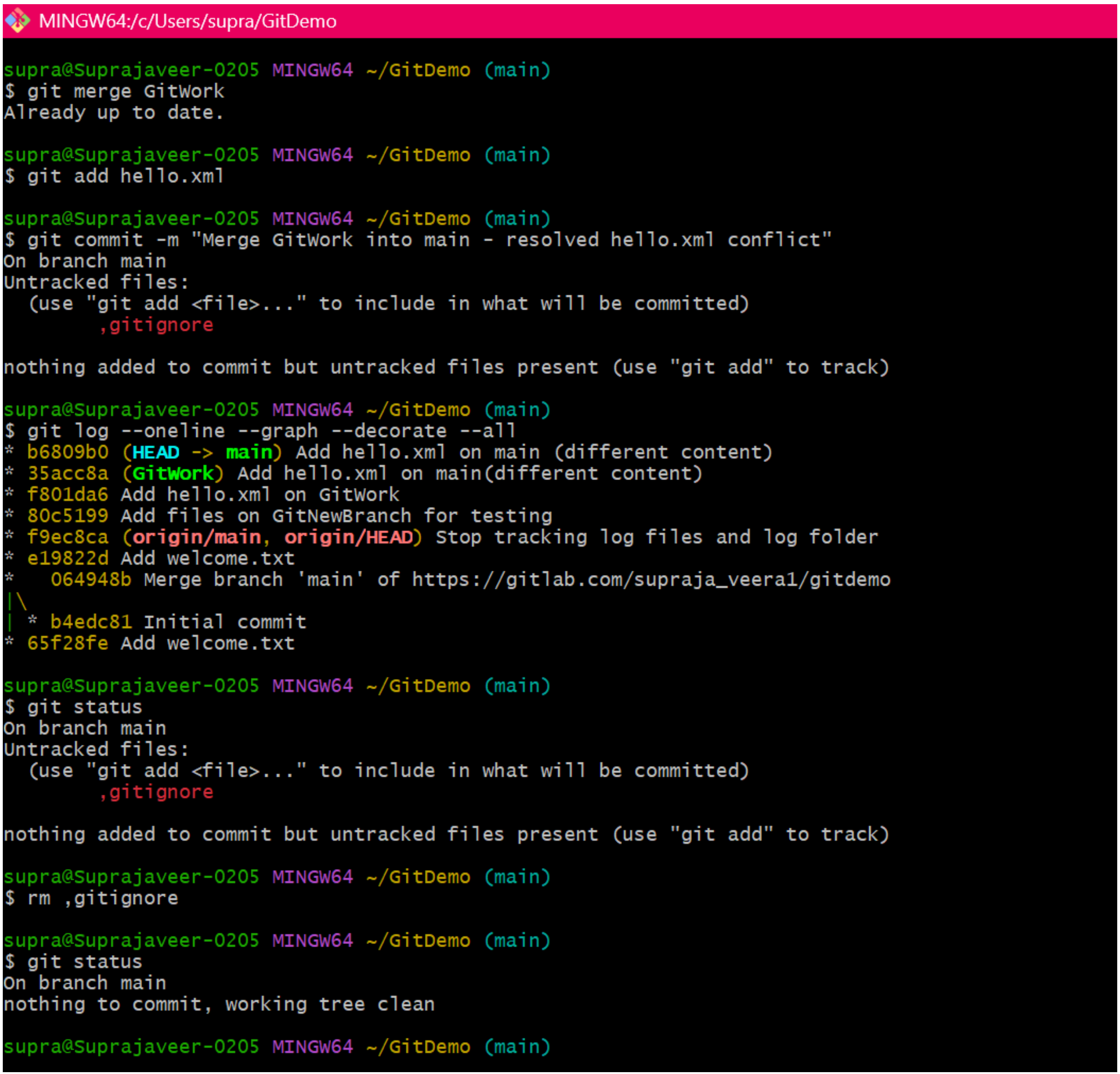
Please follow the instructions to complete the hands-on. Each instruction expect a command for the Git Bash.

1. Verify if master is in clean state.
2. Create a branch **"GitWork".** Add a file "hello.xml".
3. Update the content of "hello.xml" and observe the status
4. Commit the changes to reflect in the branch
5. Switch to master.
6. Add a file **"hello.xml"** to the master and add some different content than previous.
7. Commit the changes to the master
8. Observe the log by executing **"git log –oneline –graph –decorate –all"**
9. Check the differences with Git diff tool
10. For better visualization, use P4Merge tool to list out all the differences between master and branch
11. Merge the bran to the master
12. Observe the git mark up.
13. Use 3-way merge tool to resolve the conflict
14. Commit the changes to the master, once done with conflict
15. Observe the git status and add backup file to the .gitignore file.
16. Commit the changes to the .gitignore
17. List out all the available branches
18. Delete the branch, which merge to master.
19. Observe the log by executing **"git log –oneline –graph –decorate"**



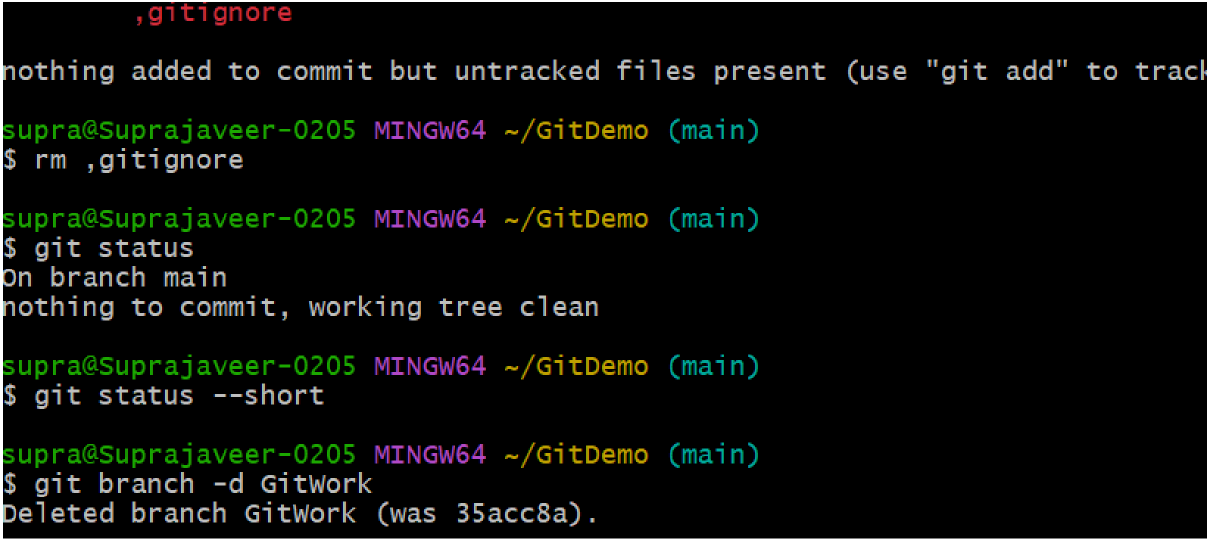
MINGW64:/c/Users/supra/GitDemo

```
supra@Suprajaveer-0205 MINGW64 ~
$ cd ~/GitDemo

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git branch
* main

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git checkout -b GitWork
Switched to a new branch 'GitWork'

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (GitWork)
$ echo "<message>Added on GitWork</message>" hello.xml
<message>Added on GitWork</message> hello.xml

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (GitWork)
$ git add hello.xml

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (GitWork)
$ git commit -m "Add hello.xml on GitWork"
On branch GitWork
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ,gitignore

nothing added to commit but untracked files present (use "git add" to track)

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (GitWork)
$ git checkout main
Switched to branch 'main'

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ echo "<hello>Main version</hello>" > hello.xml

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ echo "<message>Added on main</message>" >>hello.xml

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git add hello.xml
warning: in the working copy of 'hello.xml', LF will be replaced by CRLF the nex
t time Git touches it

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git commit -m "Add hello.xml on main (different content)"
[main b6809b0] Add hello.xml on main (different content)
 1 file changed, 1 insertion(+)
```



MINGW64:/c/Users/supra/GitDemo

```
supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git merge GitWork
Already up to date.

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git add hello.xml

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git commit -m "Merge GitWork into main - resolved hello.xml conflict"
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ,gitignore

nothing added to commit but untracked files present (use "git add" to track)

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git log --oneline --graph --decorate --all
* b6809b0 (HEAD -> main) Add hello.xml on main (different content)
* 35acc8a (GitWork) Add hello.xml on main(different content)
* f801da6 Add hello.xml on GitWork
* 80c5199 Add files on GitNewBranch for testing
* f9ec8ca (origin/main, origin/HEAD) Stop tracking log files and log folder
* e19822d Add welcome.txt
*   064948b Merge branch 'main' of https://gitlab.com/supraja_veera1/gitdemo
|\
| * b4edc81 Initial commit
* 65f28fe Add welcome.txt

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ,gitignore

nothing added to commit but untracked files present (use "git add" to track)

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ rm ,gitignore

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git status
On branch main
nothing to commit, working tree clean

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
```



```
        ,gitignore

nothing added to commit but untracked files present (use "git add" to track

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ rm ,gitignore

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git status
On branch main
nothing to commit, working tree clean
```

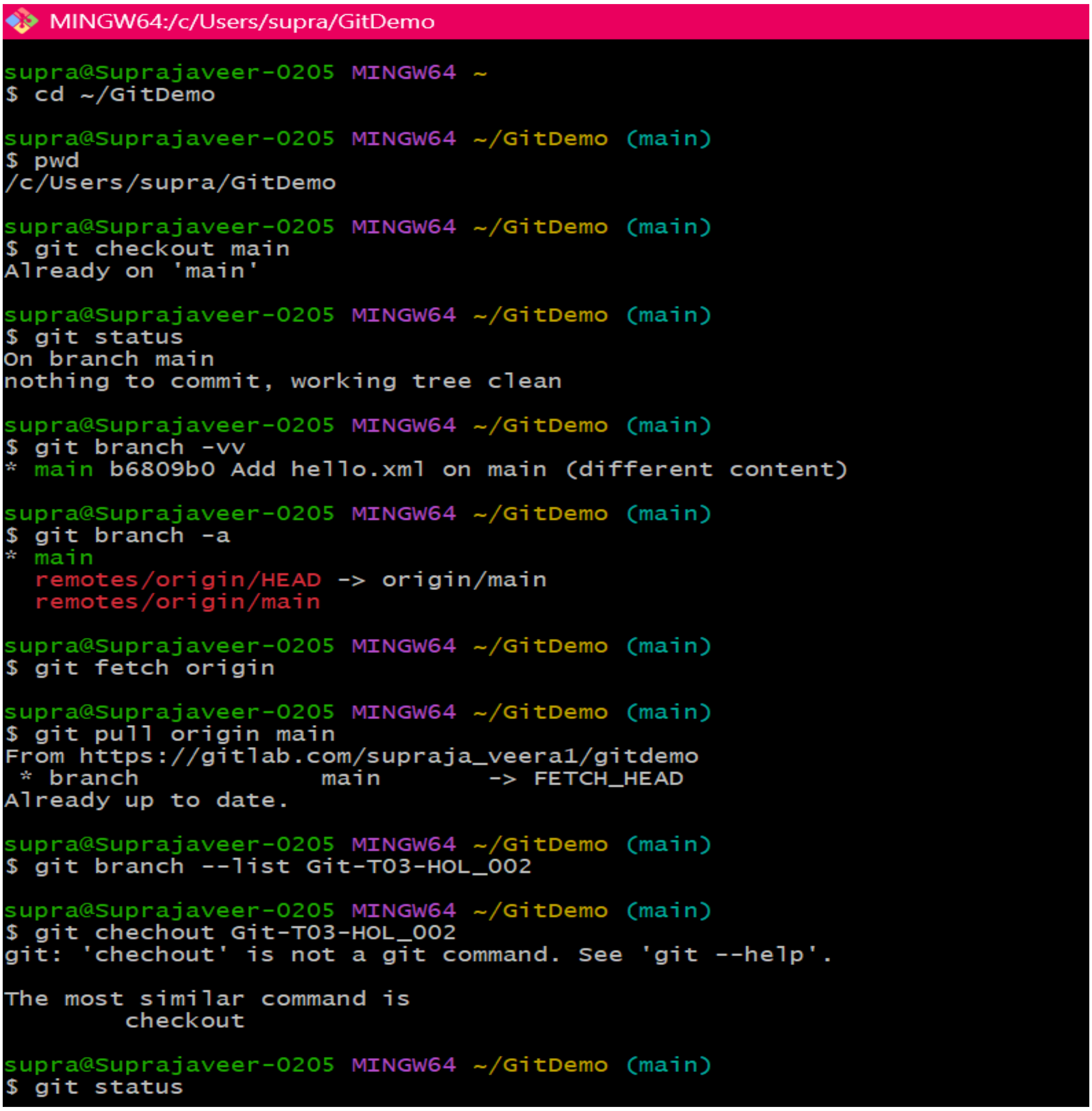
```
supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git status --short

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git branch -d GitWork
Deleted branch GitWork (was 35acc8a).
```

5.GIT-HOL

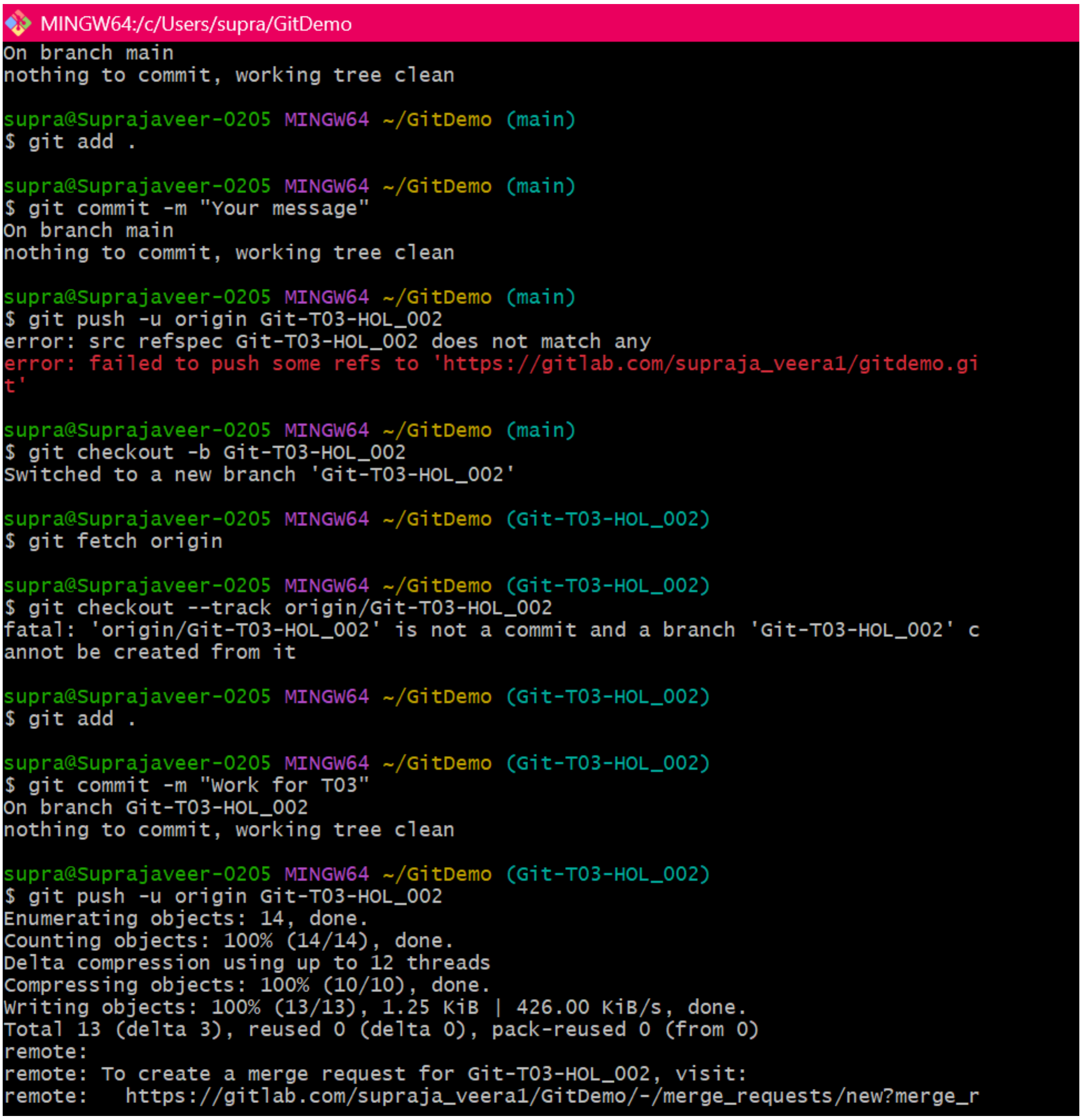
Please follow the instructions to complete the hands-on. Each instruction expects a command for the Git Bash.

1. Verify if master is in clean state.
2. List out all the available branches.
3. Pull the remote git repository to the master
4. Push the changes, which are pending from **“Git-T03-HOL_002”** to the remote repository.
5. Observe if the changes are reflected in the remote repository.

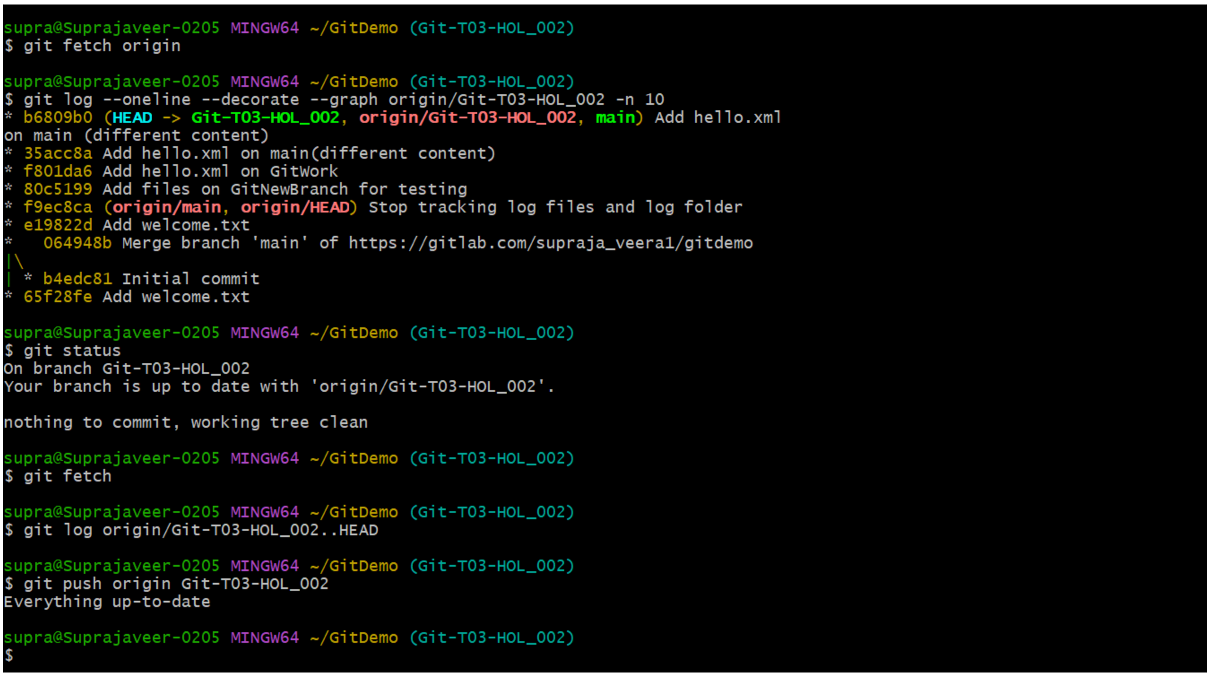
OUTPUT:



```
MINGW64:/c/Users/supra/GitDemo

supra@Suprajaveer-0205 MINGW64 ~
$ cd ~/GitDemo

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ pwd
/c/Users/supra/GitDemo

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git checkout main
Already on 'main'

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git status
On branch main
nothing to commit, working tree clean

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git branch -vv
* main b6809b0 Add hello.xml on main (different content)

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git branch -a
* main
  remotes/origin/HEAD -> origin/main
  remotes/origin/main

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git fetch origin

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git pull origin main
From https://gitlab.com/supraja_veera1/gitdemo
 * branch            main       -> FETCH_HEAD
Already up to date.

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git branch --list Git-T03-HOL_002

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git chechout Git-T03-HOL_002
git: 'chechout' is not a git command. See 'git --help'.

The most similar command is
        checkout

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git status
```



```
On branch main
nothing to commit, working tree clean

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git add .

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git commit -m "Your message"
On branch main
nothing to commit, working tree clean

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git push -u origin Git-T03-HOL_002
error: src refspec Git-T03-HOL_002 does not match any
error: failed to push some refs to 'https://gitlab.com/supraja_veera1/gitdemo.gi
t'

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (main)
$ git checkout -b Git-T03-HOL_002
Switched to a new branch 'Git-T03-HOL_002'

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (Git-T03-HOL_002)
$ git fetch origin

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (Git-T03-HOL_002)
$ git checkout --track origin/Git-T03-HOL_002
fatal: 'origin/Git-T03-HOL_002' is not a commit and a branch 'Git-T03-HOL_002' c
annot be created from it

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (Git-T03-HOL_002)
$ git add .

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (Git-T03-HOL_002)
$ git commit -m "Work for T03"
On branch Git-T03-HOL_002
nothing to commit, working tree clean

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (Git-T03-HOL_002)
$ git push -u origin Git-T03-HOL_002
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 12 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (13/13), 1.25 KiB | 426.00 KiB/s, done.
Total 13 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: To create a merge request for Git-T03-HOL_002, visit:
remote:   https://gitlab.com/supraja_veera1/GitDemo/-/merge_requests/new?merge_r
```



```
supra@Suprajaveer-0205 MINGW64 ~/GitDemo (Git-T03-HOL_002)
$ git fetch origin

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (Git-T03-HOL_002)
$ git log --oneline --decorate --graph origin/Git-T03-HOL_002 -n 10
* b6809b0 (HEAD -> Git-T03-HOL_002, origin/Git-T03-HOL_002, main) Add hello.xml
on main (different content)
* 35acc8a Add hello.xml on main(different content)
* f801da6 Add hello.xml on GitWork
* 80c5199 Add files on GitNewBranch for testing
* f9ec8ca (origin/main, origin/HEAD) Stop tracking log files and log folder
* e19822d Add welcome.txt
*   064948b Merge branch 'main' of https://gitlab.com/supraja_veera1/gitdemo
|\
| * b4edc81 Initial commit
* 65f28fe Add welcome.txt

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (Git-T03-HOL_002)
$ git status
On branch Git-T03-HOL_002
Your branch is up to date with 'origin/Git-T03-HOL_002'.

nothing to commit, working tree clean

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (Git-T03-HOL_002)
$ git fetch

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (Git-T03-HOL_002)
$ git log origin/Git-T03-HOL_002..HEAD

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (Git-T03-HOL_002)
$ git push origin Git-T03-HOL_002
Everything up-to-date

supra@Suprajaveer-0205 MINGW64 ~/GitDemo (Git-T03-HOL_002)
$
```